ELSEVIER
Linear Algebra and its Applications 339 (2001) 59–73
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

On the computational complexity of reconstructing three-dimensional lattice sets from their two-dimensional X-rays ☆

S. Brunetti [a,*], A. Del Lungo [b], Y. Gerard [c]

[a] Dipartimento di Sistemi e Inform., Università di Firenze, Via Lombroso 6/17, Firenze 50134, Italy
[b] Dipartimento di Matematica, Università di Siena, Via del Capitano 15, 53100 Siena, Italy
[c] Laboratoire de Logique et d'Informatique de Clermont-1 (LLAIC1), I.U.T. Informatique, Ensemble Universitaire des Cézeaux, B.P. No. 86, 63172 Aubière Cedex, France

Received 13 October 2000; accepted 16 July 2001

Submitted by G.T. Herman

Abstract

A generalization of a classical discrete tomography problem is considered: reconstruct three-dimensional lattice sets from their two-dimensional X-rays parallel to three coordinate planes. First, we prove that this reconstruction problem is NP-hard. Then we propose some greedy algorithms that provide approximate solutions of the problem. © 2001 Elsevier Science Inc. All rights reserved.

Keywords: Combinatorial problems; Discrete tomography; Three-dimensional lattice sets; Two-dimensional X-rays; NP-hardness; Greedy algorithm

1. Introduction

The reconstruction of binary matrices from their row and column sums is a classical problem of discrete tomography [12]. The problem has been solved by Ryser [13] and Gale [6] (independently of each other) in 1957. They gave an exact combinatorial characterization of the row and column sums that correspond to a binary matrix,

☆ This work is partially supported by MURST project: *Modelli di calcolo innovativi*: *metodi sintattici e combinatori* and by "Piano di Ateneo per la Ricerca" of the University of Siena.

* Corresponding author.
E-mail addresses: brunetti@dsi.unifi.it (S. Brunetti), dellungo@unisi.it (A. Del Lungo), gerard@llaic.u-clermont1.fr (Y. Gerard).

PII: S 0 0 2 4 - 3 7 9 5 (0 1) 0 0 4 3 7 - 2

and they derived a fast $O(nm)$ time algorithm for reconstructing a matrix from its row and column sums, where n and m denote the sizes of the matrix. We refer the reader to an excellent survey on the binary matrices with given row and column sums by Brualdi [2]. The problem is equivalent to the reconstruction of two-dimensional lattice sets from their X-rays parallel to the horizontal and vertical directions. A two-dimensional lattice set is a finite subset of the integer lattice $\mathbb{Z}^2$, and an X-ray of a lattice set parallel to a direction u is a function giving the number of its points on each line parallel to u. Many extensions of this problem have been studied. For example, it is shown that the question of reconstructing two-dimensional lattice sets from their X-rays in a set of $m \geqslant 3$ pairwise nonparallel directions is NP-hard [8]. This means that (unless $P = NP$) exact solutions of the reconstruction problem require, in general, an exponential amount of time. Polynomial algorithms have been determined to reconstruct some special sets having convexity or connectivity properties such as horizontally and vertically convex polyominoes [1,5].

By increasing the dimension, the two-dimensional reconstruction problem can be extended by several different ways and it leads in dimension 3 to two natural generalizations: the first one uses X-rays according to the lines parallel to the three axes, whereas the second uses X-rays according to the planes orthogonal to the three axes (i.e., the number of points of the lattice set on each plane orthogonal to one of the axes). The difference lies in the dimension of the linear spaces used for the X-rays but though they correspond by duality, these two extensions are not equivalent. The first generalization is a well-known NP-hard problem: Irving and Jerrum [11] proved this result in 1994; Gardner et al. [8] gave a different proof in 1999. The second one raised by Gardner and Gritzmann [7] in the book is an open problem.

The aim of this paper is to prove that this second generalization is also NP-hard. From our reduction it follows that the problem is NP-hard even in the special case where the three-dimensional lattice subsets are 6-connected and convex along the lines parallel to the three axes. We point out that these sets are the natural three-dimensional generalization of horizontally and vertically convex polyominoes.

Gritzmann et al. [9] performed a careful investigation of the performances of some algorithms for finding approximate solutions for the problem of reconstructing lattice sets from their X-rays. These algorithms yield very good worst-case bounds. Furthermore, it is shown that some greedy algorithms for one-dimensional X-rays perform even better in computational practice.

In Section 4, we use these greedy approximation algorithms for reconstructing three-dimensional lattice sets from their two-dimensional X-rays. We show that the computational results are much better than the theoretical worst-case bounds.

2. Definitions and notations

We give some formal definitions. For $k, d \in \mathbb{N}$ with $k \leqslant d - 1$, let $\mathscr{S}_{k,d}$ be the set of all k-dimensional subspaces in d-dimensional Euclidean space $\mathbb{E}^d$. Let $\mathscr{F}^d$

denote the class of all finite subsets of $\mathbb{Z}^d$ and let F be a finite subset of $\mathscr{F}^d$; we call F a *lattice set*. For $F \in \mathscr{F}^d$ let $|F|$ be the cardinality of F. Let $S \in \mathscr{S}_{k,d}$, and let $\mathscr{A}(S)$ denote the set of all k-dimensional affine subspaces of $\mathbb{E}^d$ that are parallel to S. Specially, we will consider the subspaces S_1, S_2 and S_3 of $\mathscr{S}_{2,d}$ which are orthogonal to vectors $e_1 = (1, 0, 0)$, $e_2 = (0, 1, 0)$ and $e_3 = (0, 0, 1)$, respectively, and the subspaces $S'_1 = \mathrm{lin}\,(1, 0)$, $S'_2 = \mathrm{lin}\,(0, 1)$ and $S'_3 = \mathrm{lin}\,(1, -1)$ of $\mathscr{S}_{1,d}$. The (*discrete*) *k-dimensional X-ray of F parallel to S gives the number of points of F belonging to T for* $T \in \mathscr{A}(S)$. Thus, given a subset F of $\mathscr{F}^3$ contained in $\{0, \ldots, N\}^3$, the *two-dimensional X-rays* $X_{S_1}F$, $X_{S_2}F$ and $X_{S_3}F$ of F are functions from $\mathbb{Z}$ to $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ defined by

$$X_{S_1}F_i = |\{P = (x_P, y_P, z_P) \in F \mid x_P = i\}|,$$
$$X_{S_2}F_j = |\{P = (x_P, y_P, z_P) \in F \mid y_P = j\}|,$$
$$X_{S_3}F_k = |\{P = (x_P, y_P, z_P) \in F \mid z_P = k\}|$$

for any integers i, j, k between 0 and N. For example, the two-dimensional X-rays of the three-dimensional lattice set F in Fig. 1 are: $X_{S_1}F = (2, 5, 3)$, $X_{S_2}F = (3, 3, 4)$ and $X_{S_3}F = (3, 4, 3)$.

Remark 2.1.
1. $X_{S_1}F$, $X_{S_2}F$, $X_{S_3}F$ belong to $\mathbb{N}^{N+1}$ but instead of using the indices i, j, k going from 1 to $N + 1$, we use the integers from 0 to N.
2. The sum of the coordinates of each X-ray is equal to the cardinality of F for each subset $F \subset \{0, \ldots, N\}^3$.
3. We have chosen to use the cubic lattice $\{0, \ldots, N\}^3$ because it leads to the same results as $\{0, \ldots, N_x\} \times \{0, \ldots, N_y\} \times \{0, \ldots, N_z\}$ with lighter notations.

We can now introduce the generalization of the two-dimensional reconstruction problem. Let $\mathscr{E}^3$ be a subclass of three-dimensional lattice sets (i.e., $\mathscr{E}^3 \subseteq \mathscr{F}^3$). The problem consists in determining whether there exists any set F of $\mathscr{E}^3$ whose X-rays parallel to S_1, S_2 and S_3 are equal to given vectors X, Y and Z of $\mathbb{N}^n$.

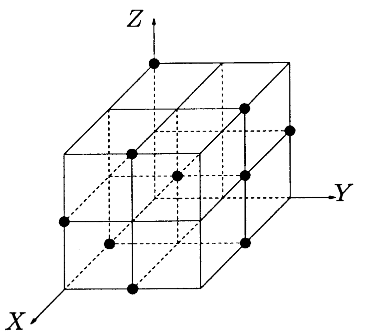

Fig. 1. A three-dimensional lattice set.

CONSISTENCY$_{\mathscr{E}^3}(S_1, S_2, S_3)$
Instance: Three vectors $X \in \mathbb{N}^{N+1}$, $Y \in \mathbb{N}^{N+1}$ and $Z \in \mathbb{N}^{N+1}$ such that $\sum_{i=0}^{N} X_i = \sum_{j=0}^{N} Y_j = \sum_{k=0}^{N} Z_k$.
Question: Does there exist a set $F \subset \{0, \ldots, N\}^3$ of $\mathscr{E}^3$ such that $X_{S_1} F = X$, $X_{S_2} F = Y$ and $X_{S_3} F = Z$?

3. NP-completeness of CONSISTENCY$_{\mathscr{F}^3}(S_1, S_2, S_3)$

The purpose of this section is to show that CONSISTENCY$_{\mathscr{F}^3}(S_1, S_2, S_3)$ is NP-complete. The reduction is done from a two-dimensional reconstruction problem. We introduce some notations which allow us to define this problem. Given a two-dimensional lattice set F' of $\{0, \ldots, n\}^2$, the *one-dimensional X-rays* $X_{S'_1} F'$, $X_{S'_2} F'$ and $X_{S'_3} F'$ of F' are defined by

$$\begin{aligned}
X_{S'_1} F'_i &= |\{P = (x_P, y_P) \in F' \mid x_P = i\}|,\\
X_{S'_2} F'_j &= |\{P = (x_P, y_P) \in F' \mid y_P = j\}|,\\
X_{S'_3} F'_k &= |\{P = (x_P, y_P) \in F' \mid x_P + y_P = k\}|
\end{aligned}$$

for integers i, j between 0 and n and for an integer k between 0 and $2n$. If $X_{S'_1} F'_i = 1$ and $X_{S'_2} F'_j = 1$ for each i, j between 0 and n, then F' is a *permutation lattice set* of size n. It is easy to prove that $\sum_{k=0}^{2n} X_{S'_3} F'_k = n + 1$ and $\sum_{k=0}^{2n} k X_{S'_3} F'_k = n(n + 1)$. The lattice set in Fig. 2 is a permutation lattice set of size 6, its X-ray parallel to the diagonal direction is $X_{S'_3} F' = (0, 0, 1, 1, 0, 2, 0, 1, 0, 0, 2, 0, 0)$, and $\sum_{k=0}^{12} X_{S'_3} F'_k = 7$, $\sum_{k=0}^{12} k X_{S'_3} F'_k = 42$. Notice that every permutation lattice set of size n corresponds to a permutation of the numbers from 0 to n. We can now define the problem:

RESTRICTED CONSISTENCY$_{\mathscr{F}^2}(S'_1, S'_2, S'_3)$
Instance: A vector $d \in \mathbb{N}^{2n+1}$ such that $\sum_{k=0}^{2n} d_k = n + 1$ and $\sum_{k=0}^{2n} k \cdot d_k = n(n + 1)$.

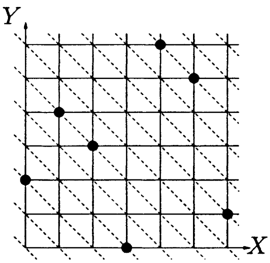

Fig. 2. A permutation lattice set.

Question: Does there exist a permutation lattice set F' of size n such that $X_{S_3'}F' = d$?

This restricted version of the two-dimensional reconstruction problem is NP-complete (see [4]). We are now in a position to prove the NP-completeness of our problem.

Theorem 3.1. *CONSISTENCY*$_{\mathscr{F}^3}(S_1, S_2, S_3)$ *is NP-complete.*

Proof. Membership in NP is immediate, since we can verify in polynomial time whether a given set F is or is not a solution.

To prove that CONSISTENCY$_{\mathscr{F}^3}(S_1, S_2, S_3)$ is NP-complete, we describe a polynomial-time transformation from RESTRICTED CONSISTENCY$_{\mathscr{F}^2}(S_1', S_2', S_3')$. Let d be an arbitrary instance of this problem. We must construct an instance X, Y, Z of CONSISTENCY$_{\mathscr{F}^3}(S_1, S_2, S_3)$ which admits a three-dimensional lattice set F such that $X_{S_1}F = X$, $X_{S_2} = Y$ and $X_{S_3} = Z$ if and only if there is a permutation lattice set F' such that $X_{S_3'}F' = d$.

Let $N = 2n$ and we introduce the discrete 3-simplex P defined by $P = \{(i, j, k) \in \{0, \ldots, 2n\}^3 \mid i + j + k < 2n\}$. The three two-dimensional X-rays of P are $X_{S_1}P$, $X_{S_2}P$ and $X_{S_3}P$. The instance X, Y, Z of CONSISTENCY$_{\mathscr{F}^3}(S_1, S_2, S_3)$ is defined in the following way (see Fig. 3):

- $X_i = X_{S_1}P_i + 1$ for the indices $i \in \{0, \ldots, n\}$,
  $X_{i'} = X_{S_1}P_{i'}$ for the indices $i' \in \{n + 1, \ldots, 2n\}$,

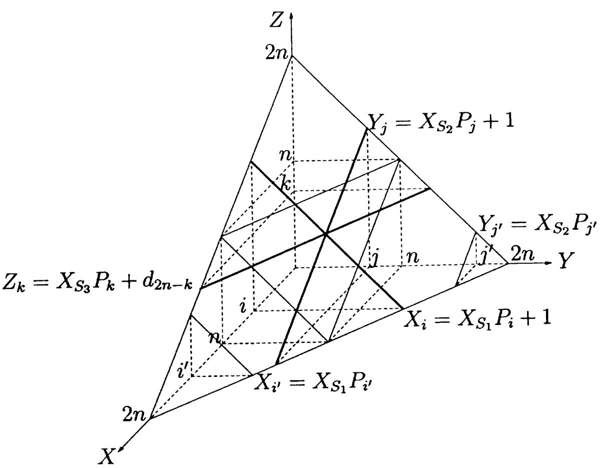

Fig. 3. The instance X, Y, Z of CONSISTENCY$_{\mathscr{F}^3}(S_1, S_2, S_3)$ obtained by an arbitrary instance d of RESTRICTED CONSISTENCY$_{\mathscr{F}^2}(S_1', S_2', S_3')$.

- $Y_j = X_{S_2} P_j + 1$ for the indices $j \in \{0, \ldots, n\}$,
  $Y_{j'} = X_{S_2} P_{j'}$ for the indices $j' \in \{n+1, \ldots, 2n\}$,
- $Z_k = X_{S_3} P_k + d_{2n-k}$ for the indices $k \in \{0, \ldots, 2n\}$.

Notice that the intersections between the plane $x + y + z = 2n$ and the planes $x = i$, $y = j$ and $z = k$, with $i, j, k = 0, 1, \ldots, 2n$, provide the set of points on the surface of the simplex P which is isomorphic to a two-dimensional lattice (see Fig. 4). So, roughly speaking, the basic idea of the transformation is to embed the permutation lattice set F' in the surface of P (see Fig. 4).

We now have to establish that the derived instance X, Y, Z admits a lattice set F such that $X_{S_1} F = X$, $X_{S_2} = Y$ and $X_{S_3} = Z$ if and only if there is a permutation lattice set F' such that $X_{S'_3} F' = d$.

Suppose first that F' is a permutation lattice set such that $X_{S'_3} F' = d$. From the definition of X, Y, Z it follows that the three-dimensional lattice set $F \subset \{0, \ldots, 2n\}^3$ defined by

$$F = P \cup \left\{(i, j, k) \in \{0, \ldots, n\}^3 \mid i + j + k = 2n,\ (i, j) \in F'\right\}$$

verifies $X_{S_1} F = X$, $X_{S_2} F = Y$ and $X_{S_3} F = Z$ (see Fig. 4).

On the other hand, suppose that F is a three-dimensional lattice set such that $X_{S_1} F = X$, $X_{S_2} F = Y$ and $X_{S_3} F = Z$. If we are able to prove that

$$\begin{aligned} &\left\{(i, j, k) \in \{0, \ldots, 2n\}^3 \mid i + j + k < 2n\right\} \\ &\quad \subset F \subset \left\{(i, j, k) \in \{0, \ldots, 2n\}^3 \mid i + j + k \leqslant 2n\right\}, \end{aligned} \tag{1}$$

then by the definition of X, Y, Z, the set F is the union of P with a set $H = \{(i, j, k) \in \{0, \ldots, n\}^3 \mid i + j + k = 2n,\ (i, j) \in F'\}$, where F' is a permutation lattice set such that $X_{S'_3} F' = d$ (see Fig. 4).

Let A' be a permutation lattice set. The three-dimensional lattice set $A \subset \{0, \ldots, 2n\}^3$ defined by

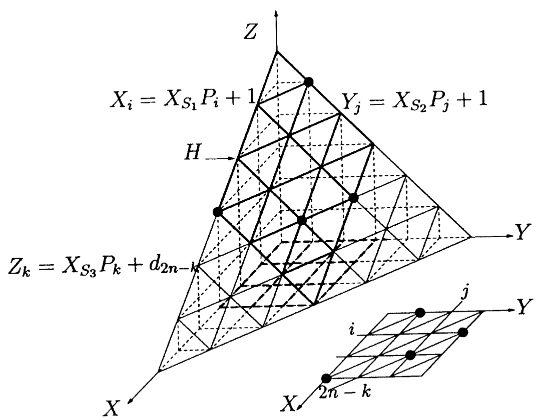

Fig. 4. A permutation lattice set and its corresponding three-dimensional lattice set.

$$A = P \cup \left\{(i, j, k) \in \{0, \dots, n\}^3 \mid i + j + k = 2n,\ (i, j) \in A'\right\}$$

verifies $X_{S_1}A = X$ and $X_{S_2}A = Y$. Let us take two properties of the barycenter G_A of A into consideration. For any set $S \subset \{0, \dots, 2n\}^3$ having the same number of points of A,

$$\overrightarrow{OG}_S.\vec{u} \geqslant \overrightarrow{OG}_A \cdot \vec{u},$$

where $\vec{u}$ is the vector of coordinates $(1, 1, 1)$, O is the origin of $\mathbb{R}^3$ and G_S is the barycenter of S. Moreover, we can prove that if $\overrightarrow{OG}_S \cdot \vec{u} = \overrightarrow{OG}_A \cdot \vec{u}$, then

$$\begin{aligned}&\left\{(i, j, k) \in \{0, \dots, 2n\}^3 \mid i + j + k < 2n\right\}\\ &\quad \subset S \subset \left\{(i, j, k) \in \{0, \dots, 2n\}^3 \mid i + j + k \leqslant 2n\right\}.\end{aligned}$$

The coordinates of G_A are

$$x_{G_A} = \frac{\left(\sum_{i=0}^{2n} i\ X_i\right)}{|A|}, \quad y_{G_A} = \frac{\left(\sum_{j=0}^{2n} j\ Y_j\right)}{|A|}, \quad z_{G_A} = \frac{\left(\sum_{k=0}^{2n} k\ X_{S_3}A_k\right)}{|A|}.$$

From the definition of A it follows that

$$\begin{aligned}z_{G_A} = \Bigg(\sum_{k=0}^{2n} k\ X_{S_3}P_k + k|\,\{(i, j) \in \{0, \dots, n\}^2 \mid i + j + k = 2n,\\ (i, j) \in A'\}|\Bigg)/|A|,\end{aligned}$$

and so

$$z_{G_A} = \frac{\left(\sum_{k=0}^{2n} k\ X_{S_3}P_k + k\ X_{S_3'}A_k'\right)}{|A|}.$$

Since A' is a permutation lattice set, we have that $\sum_{k=0}^{2n} kX_{S_3'}A_k' = n(n+1)$. Therefore,

$$z_{G_A} = \frac{\left(\sum_{k=0}^{2n} k\ X_{S_3}P_k + n(n+1)\right)}{|A|}.$$

The coordinates of the barycenter G_F of F are

$$x_{G_F} = \frac{\left(\sum_{i=0}^{2n} i\ X_i\right)}{|F|}, \quad y_{G_F} = \frac{\left(\sum_{j=0}^{2n} j\ Y_j\right)}{|F|}, \quad z_{G_F} = \frac{\left(\sum_{k=0}^{2n} k\ Z_k\right)}{|F|}.$$

By $Z_k = X_{S_3}P_k + d_{2n-k}$, the height is such that

$$z_{G_F} = \frac{\left(\sum_{k=0}^{2n} k\ X_{S_3}P_k + k\ d_{2n-k}\right)}{|F|}.$$

Since $\sum_{k=0}^{2n} k \cdot d_k = n(n+1)$, we have that $\sum_{k=0}^{2n} k \cdot d_{2n-k} = n(n+1)$. Therefore, $z_{G_F} = (\sum_{k=0}^{2n} k\ X_{S_3}P_k + n(n+1)/|F|$. Finally, since $|F| = |A|$, we deduce that

$G_F = G_A$. By using the properties of the barycenter G_A, we obtain that F satisfies condition (1), and the theorem is proved. □

In some practical applications we have some a priori knowledge about the sets to be reconstructed. The algorithms can take advantage of this information to reconstruct the set. Mathematically, these properties can be described in terms of a subclass of two-dimensional lattice sets to which the solution must belong. For instance, there are polynomial-time algorithms to reconstruct horizontally and vertically convex polyominoes (i.e., two-dimensional lattice subsets which are 4-connected and convex in the horizontal and vertical directions) from their X-rays in horizontal and vertical directions [1,5]. Unfortunately, from the proof of Theorem 3.1, it follows that our three-dimensional reconstruction problem is NP-complete on many subclasses of three-dimensional lattice sets.

Corollary 3.1. *CONSISTENCY*$_{\mathscr{E}^3}(S_1, S_2, S_3)$ *is NP-complete for each subclass* $\mathscr{E}^3$ *of three-dimensional lattice sets containing all the lattice sets F of* $\{0, \dots, N\}^3$ *such that*

$$\begin{aligned} &\left\{(i, j, k) \in \{0, \dots, N\}^3 \mid i + j + k < N\right\} \\ &\quad \subset F \subset \left\{(i, j, k) \in \{0, \dots, N\}^3 \mid i + j + k < N + 1\right\}. \end{aligned} \tag{2}$$

This corollary has many consequences as the NP-completeness of the problem on the classes of 6-, 18-, 26-connected lattice sets or with the ones whose sets are convex according to directions $(t, u, v) \in \mathbb{R}^3 \backslash \{(0, 0, 0)\}$ verifying $t + u + v \neq 0$. In particular, the problem is NP-complete on the class of three-dimensional lattice subsets which are 6-connected and convex along the directions e_1, e_2 and e_3, namely, the natural three-dimensional generalization of horizontally and vertically convex polyominoes. A natural class in which it seems possible to have a polynomial result is the one of three-dimensional convex lattice sets. In fact, a three-dimensional convex lattice set is a set $F \in \mathscr{F}^3$ such that $F = \mathbb{Z}^3 \cap \operatorname{conv} F$, and so this class does not satisfy condition (2) of Corollary 3.1. However, the complexity of the problem of reconstructing two-dimensional convex lattice sets from their X-rays in the horizontal and vertical directions is open. The reconstruction problem on this class can be solved in polynomial time if we take the X-rays in some sets of four prescribed lattice directions or in any set of seven prescribed mutually nonparallel lattice directions [3].

Irving and Jerrum [11] proved that the reconstruction of three-dimensional sets from their one-dimensional X-rays parallel to e_1, e_2 and e_3 is NP-complete. We wish to point out a case in which the reconstruction problems with one- and two-dimensional X-rays are not equivalent from a computational complexity point of view. Irving and Jerrum show that the problem with one-dimensional X-rays parallel to e_1, e_2 and e_3 is NP-complete, even in the special case where all the elements of the X-rays are 0 or 1. On the contrary, CONSISTENCY$_{\mathscr{E}^3}(S_1, S_2, S_3)$ can be solved in polynomial time in the special case where all the elements of the X-rays are 0 or 1. In fact, it is easy to prove that:

Proposition 3.1. *Let $X \in \{0, 1\}^{N+1}$, $Y \in \{0, 1\}^{N+1}$ and $Z \in \{0, 1\}^{N+1}$. There exists at least one lattice set $F \subset \{0, \dots, N\}^3$ such that $X_{S_1}F = X$, $X_{S_2} = Y$ and $X_{S_3} = Z$ if and only if*

$$\sum_{i=0}^{N} X_i = \sum_{j=0}^{N} Y_j = \sum_{k=0}^{N} Z_k.$$

4. Approximation algorithms

There is a solution for a given instance of CONSISTENCY$_{\mathscr{F}^3}(S_1, S_2, S_3)$ if the corresponding question has an affirmative answer. From a practical point of view, the most relevant task is that of reconstructing this solution. The reconstruction problem RECONSTRUCTION$_{\mathscr{F}^3}(S_1, S_2, S_3)$ is defined in a way similar to CONSISTENCY$_{\mathscr{F}^3}(S_1, S_2, S_3)$, the input being the same but the question replaced by the task of constructing a solution if one exists. By the result of the previous section, RECONSTRUCTION$_{\mathscr{F}^3}(S_1, S_2, S_3)$ is NP-hard. This means that (unless P = NP) exact solutions of the problem require, in general, an exponential amount of time. In polynomial time only approximate solutions can be expected. In this context, an approximate solution is close to the optimal one if its X-rays parallel to S_1, S_2 and S_3 are close to the given input data.

Gritzmann et al. [9] presented some greedy algorithms for finding approximate solutions for the problem of reconstructing lattice sets from their X-rays. These algorithms yield very good worst-case bounds. Furthermore, it is shown that by performing these algorithms for reconstructing two-dimensional lattice sets from their one-dimensional X-rays parallel to a set of m directions, with $3 \leqslant m \leqslant 5$, the computational results are even better than the theoretical worst-case bounds. The aim of this section is to perform these greedy algorithms for reconstructing three-dimensional lattice sets from their two-dimensional X-rays parallel to S_1, S_2 and S_3. First, we describe the results established by Gritzmann et al., then the performance of these greedy algorithms on our problem. Let us take the following optimization problem into consideration:

BEST-INNER-FIT$_{\mathscr{F}^3}(S_1, S_2, S_3)$ [BIF]
Instance: Three vectors $X \in \mathbb{N}^{N+1}$, $Y \in \mathbb{N}^{N+1}$ and $Z \in \mathbb{N}^{N+1}$ such that $\sum_{i=0}^{N} X_i = \sum_{j=0}^{N} Y_j = \sum_{k=0}^{N} Z_k$.
Question: Find a three-dimensional lattice set $F \subset \{0, \dots, N\}^3$ of maximal cardinality such that $X_{S_1}F_i \leqslant X_i$, $X_{S_2}F_j \leqslant Y_j$ and $X_{S_3}F_k \leqslant Z_k$ for all $i, j, k = 0, \dots, N$.

We use three greedy algorithms proposed in [9] for solving [BIF]. The first one (Greedy A) is a simple greedy procedure which considers all positions of $\{0, \ldots, N\}^3$ in a random order and tries to insert points at these positions.

procedure *Greedy A*
Calculate a random permutation of all points of $\{0, \ldots, N\}^3$
For each point in the order of this permutation **do**
    Check whether any plane passing through this point is saturated
    If no plane is saturated **then**
        Add the point to the solution set
        Update the sums of the planes passing through this point.

A smarter way to insert points into the solution set is to apply *back-projection technique*, where each candidate point has a weight based on the X-ray values of the three planes through this point. The following two algorithms perform this technique.

Let P be a point of $\{0, \ldots, N\}^3$ and let Π be the plane through this point and parallel to S_i. The weight $W_i(P)$ of P is the ratio between the number of points still to be inserted on Π and the number of candidate points still available on Π. If P is a candidate point and $W_i(P) = 0$, then P cannot be inserted into the solution set. On the contrary, if $W_i(P) = 1$, then P must be inserted into the solution set.

The second algorithm (Greedy B) is a variant of the second one proposed by Gritzmann et al. [9]. It sorts vector $Z = (Z_0, \ldots, Z_N)$ by decreasing order and considers the corresponding planes parallel to S_3 in this decreasing plane-weight order. This is the order in which candidate points of $\{0, \ldots, N\}^3$ are considered for insertion into the solution set. For a fixed plane Π parallel to S_3, each point P of Π gets the weight $W_1(P)W_2(P)$. This product is a natural indicator for comparing the importance of the candidate points on Π. The algorithm finds the point of maximum weight and tries to insert this point into the solution set. If the point is inserted, the algorithm updates the weights of the points of Π and repeats the insertion procedure. We use a heap for keeping the points ordered according to their weights.

procedure *Greedy B*
Sort the planes parallel to S_3 by decreasing plane-weights
For each of these planes (Π) in this order **do**
    For each point P on Π **do**
        Calculate its weight $W_1(P)W_2(P)$ and insert it into the heap
    While there are still points in the heap **do**
        Find the maximum weight and corresponding point and remove it from the heap
        Check whether any plane passing through this point is saturated
        If no plane is saturated **then**
            Add the point to the solution set
            Update the sums of the planes passing through this point.

Let us stress the fact that Greedy B is the natural extension of the algorithm for reconstructing two-dimensional lattice sets from their X-rays parallel to the horizontal and vertical directions defined by Ryser in [13].

The third algorithm (Greedy C) computes the weight $W_1(P)W_2(P)W_3(P)$ for each point P of $\{0, \ldots, N\}^3$. It finds the point of maximum weight and tries to insert this point into the solution set. If the point is inserted, it updates the weights of the points and repeats the insertion procedure.

procedure *Greedy C*
For each point P **do**
  Calculate its weight $W_1(P)W_2(P)W_3(P)$ and insert it into the heap
While there are still points in the heap **do**
  Find the maximum weight and corresponding point and remove it from the heap
  Check whether any plane passing through this point is saturated
  If no plane is saturated **then**
    Add the point to the solution set
    Update the sums of the planes passing through this point.

Greedy A, Greedy B and Greedy C provide a three-dimensional lattice set V such that $|V|/|F| \leqslant 1$, where F is an optimal solution of [BIF]. Gritzmann et al. [9] give a theoretical worst-case lower bound of $|V|/|F|$ for a wide class of algorithms that fit into a general paradigm. The previous greedy algorithms belong to this class and their lower bound is

$$|V|/|F| \geqslant 1/3.$$

We performed the algorithms for problems of size $5 \times 5 \times 5$ to $50 \times 50 \times 50$ and density of the set between 5% and 50%. We consider the average performances of randomly instances, where all the instances are consistent. The computational results are much better than the theoretical worst-case bound. Tables 1, 2 and 3 show the performances of Greedy A on instances of 5%, 10% and 50% density, respectively. Tables 4, 5 and 6 show the performances of Greedy B on instances of 5%, 10% and 50% density, respectively. The column “Optimum found” gives the number of tests in which the algorithm constructs a lattice set having X-rays equal to the vectors of the instance. The running-times are in milliseconds.

The computational study shows that Greedy A has small running-times and small errors. It is hard to think of any algorithm that is simpler than Greedy A, and so it is a wonder that it provides small errors. The ratio of the cardinality of the approximate solution determined by Greedy A to that an optimal solution is: $0.988947 \leqslant |V|/|F| \leqslant 0.997369$, $0.979480 \leqslant |V|/|F| \leqslant 0.997701$ and $0.965471 \leqslant |V|/|F| \leqslant 0.998363$, for instances of 5%, 10% and 50% density, respectively.

The weights assigned dynamically to the candidate points by Greedy C provide very small errors and very long running-times. We do not report the performances

Table 1
Performances of Greedy A on instances of 5% density

Problem size	No. of examples	Optimum found	$\|V\|/\|F\|$	Running-times
$5 \times 5 \times 5$	4000	3705	0.988947	0
$10 \times 10 \times 10$	4000	2357	0.986953	1
$15 \times 15 \times 15$	4000	1223	0.989723	2
$20 \times 20 \times 20$	4000	614	0.992055	4
$25 \times 25 \times 25$	4000	317	0.993880	10
$30 \times 30 \times 30$	4000	159	0.995113	20
$35 \times 35 \times 35$	2000	28	0.995874	35
$40 \times 40 \times 40$	2000	20	0.996528	60
$45 \times 45 \times 45$	1000	8	0.997010	101
$50 \times 50 \times 50$	1000	2	0.997369	137

Table 2
Performances of Greedy A on instances of 10% density

Problem size	No. of examples	Optimum found	$\|V\|/\|F\|$	Running-times
$5 \times 5 \times 5$	4000	3146	0.979480	0
$10 \times 10 \times 10$	4000	1228	0.983290	1
$15 \times 15 \times 15$	4000	419	0.988891	3
$20 \times 20 \times 20$	4000	166	0.992251	8
$25 \times 25 \times 25$	4000	57	0.994069	16
$30 \times 30 \times 30$	4000	21	0.995351	31
$35 \times 35 \times 35$	2000	4	0.996253	56
$40 \times 40 \times 40$	2000	2	0.996253	93
$45 \times 45 \times 45$	1000	0	0.997319	155
$50 \times 50 \times 50$	1000	0	0.997701	233

Table 3
Performances of Greedy A on instances of 50% density

Problem size	No. of examples	Optimum found	$\|V\|/\|F\|$	Running-times
$5 \times 5 \times 5$	4000	634	0.965471	0
$10 \times 10 \times 10$	4000	29	0.984323	3
$15 \times 15 \times 15$	4000	2	0.990769	12
$20 \times 20 \times 20$	4000	0	0.993827	32
$25 \times 25 \times 25$	4000	0	0.995495	68
$30 \times 30 \times 30$	4000	0	0.996533	144
$35 \times 35 \times 35$	2000	0	0.997269	239
$40 \times 40 \times 40$	2000	0	0.997753	389
$45 \times 45 \times 45$	1000	0	0.998098	613
$50 \times 50 \times 50$	1000	0	0.998363	865

Table 4
Performances of Greedy B on instances of 5% density

Problem size	No. of examples	Optimum found	$\lvert V\rvert/\lvert F\rvert$	Running-times
$5 \times 5 \times 5$	4000	4000	1	0.3
$10 \times 10 \times 10$	4000	4000	1	2
$15 \times 15 \times 15$	4000	4000	1	13
$20 \times 20 \times 20$	4000	4000	1	35
$25 \times 25 \times 25$	4000	4000	1	101
$30 \times 30 \times 30$	2000	2000	1	311
$35 \times 35 \times 35$	2000	2000	1	630
$40 \times 40 \times 40$	2000	2000	1	1185
$45 \times 45 \times 45$	1000	1000	1	2184
$50 \times 50 \times 50$	1000	1000	1	4312

Table 5
Performances of Greedy B on instances of 10% density

Problem size	No. of examples	Optimum found	$\lvert V\rvert/\lvert F\rvert$	Running-times
$5 \times 5 \times 5$	4000	4000	1	0.4
$10 \times 10 \times 10$	4000	4000	1	4
$15 \times 15 \times 15$	4000	4000	1	15
$20 \times 20 \times 20$	4000	4000	1	94
$25 \times 25 \times 25$	4000	4000	1	220
$30 \times 30 \times 30$	4000	4000	1	569
$35 \times 35 \times 35$	2000	2000	1	1077
$40 \times 40 \times 40$	2000	2000	1	2302
$45 \times 45 \times 45$	1000	1000	1	4404
$50 \times 50 \times 50$	1000	1000	1	8136

Table 6
Performances of Greedy B on instances of 50% density

Problem size	No. of examples	Optimum found	$\lvert V\rvert/\lvert F\rvert$	Running-times
$5 \times 5 \times 5$	4000	3996	0.997872	1
$10 \times 10 \times 10$	4000	3988	0.998935	16
$15 \times 15 \times 15$	4000	3984	0.999210	87
$20 \times 20 \times 20$	4000	3981	0.999775	355
$25 \times 25 \times 25$	4000	4000	0.999893	970
$30 \times 30 \times 30$	4000	4000	0.999900	2344
$35 \times 35 \times 35$	2000	1987	0.999918	5402
$40 \times 40 \times 40$	2000	1979	0.999935	9898
$45 \times 45 \times 45$	1000	996	0.999949	17569
$50 \times 50 \times 50$	1000	992	0.999982	33109

of Greedy C because according to our experiments, in all the studied cases, the algorithm found the optimum.

Finally, Greedy B keeps to the happy medium, since it gives small running-times and errors near to Greedy C. Since Greedy B gives an excellent performance, we do not use any improvements of the algorithms proposed in [9]. Therefore, in conclusion, the natural extension of Ryser's algorithm gives a very good procedure for finding approximate solutions for the problem of reconstructing three-dimensional lattice sets from their two-dimensional X-rays.

5. Conclusion

The most important idea presented in this paper is that instances of the problem of reconstructing a permutation lattice set from its X-ray parallel to the diagonal direction can be formulated as instances of the problem of reconstructing a three-dimensional lattice set from its two-dimensional X-rays parallel to the coordinate axes. Therefore, the NP-completeness of the former problem implies the NP-completeness of the latter problem. This result allows us to work out an open problem raised by Gardner and Gritzmann in [7].

Let us point out that this reformulation method actually applies to the reconstruction problem of any lattice set in $\{0, \ldots, n\}^2$ from its X-rays parallel to the horizontal, vertical and diagonal directions, as a referee noticed. Recently, we were informed that the approach suggested by the referee has been used to generalize our result to higher dimensions [10].

Acknowledgements

The authors would like to thank Luana Breccola for implementing the algorithms, S. de Vries for useful comments on the first preliminary version of this paper, and the referee for pointing out that our transformation can be applied to any lattice set in $\{0, \ldots, n\}^2$.

References

[1] E. Barcucci, A. Del Lungo, M. Nivat, R. Pinzani, Reconstructing convex polyominoes from their horizontal and vertical projections, Theoret. Comput. Sci. 155 (1996) 321–347.

[2] R.A. Brualdi, Matrices of zeros and ones with fixed row and column sum vectors, Linear Algebra Appl. 33 (1980) 159–231.

[3] S. Brunetti, A. Daurat, Reconstruction of discrete sets from two or more projections in any direction, in: Proceedings of the Seventh International Workshop on Combinatorial Image Analysis (IWCIA 2000), Caen, 2000, pp. 241–258.

[4] S. Brunetti, A. Del Lungo, P. Gritzmann, S. de Vries, On the reconstruction of permutation and partition sets under tomographic constraints, submitted.
[5] M. Chrobak, C. Dürr, Reconstructing hv-convex polyominoes from orthogonal projections, Inform. Process. Lett. 6 (69) (1999) 283–289.
[6] D. Gale, A theorem on flows in networks, Pacific J. Math. 7 (1957) 1073–1082.
[7] R.J. Gardner, P. Gritzmann, Uniqueness and complexity in discrete tomography, in: G.T. Herman, A. Kuba (Eds.), Discrete Tomography: Foundations Algorithms and Applications, Birkhäuser, Boston, MA, 1999, pp. 85–113.
[8] R.J. Gardner, P. Gritzmann, D. Prangenberg, On the computational complexity of reconstructing lattice sets from their X-rays, Discrete Math. 202 (1999) 45–71.
[9] P. Gritzmann, S. de Vries, M. Wiegelmann, Approximating binary images from discrete X-rays, SIAM J. Optim. 11 (2) (2000) 522–546.
[10] P. Gritzmann, S. de Vries, On the algorithm inversion of the discrete Radon transform, Private communication.
[11] R.W. Irving, M.R. Jerrum, Three-dimensional statistical data security problems, SIAM J. Comput. 23 (1994) 170–184.
[12] A. Kuba, G.T. Herman, Discrete tomography: a historical overview, in: G.T. Herman, A. Kuba (Eds.), Discrete Tomography: Foundations, Algorithms and Applications, Birkhäuser, Boston, MA, 1999, pp. 3–34.
[13] H.J. Ryser, Combinatorial properties of matrices of zeros and ones, Canad. J. Math. 9 (1957) 371–377.